# **Experiment no.1**

**AIM:** - Data Collection- YouTube, connect to and capture social media data for business (scraping, crawling, parsing).

**Theory:** In the digital era, businesses rely on social media analytics to understand audience behavior, trends, and engagement. YouTube, being one of the largest video-sharing platforms, provides valuable data, including video metadata, comments, views, likes, and subscriber counts.

## **Concepts:**

### 1. Web Scraping

- o Extracts structured data from web pages using automated scripts.
- o Tools: BeautifulSoup, Selenium, Scrapy.

## 2. Web Crawling

- o Systematically browses the web to collect data from multiple pages.
- o Tools: Scrapy, Selenium, requests.

#### 3. Parsing

- o Processes raw data into structured information.
- o JSON, XML, and HTML parsing techniques are used.

#### **YouTube Data Collection Methods:**

### 1. Using YouTube Data API

- o Official method, provides structured data but has rate limits.
- Requires API key and authentication.

# 2. Web Scraping YouTube

- o Extracts data directly from YouTube's web interface.
- o Helps bypass API limits but may face challenges due to dynamic content loading.

# **Applications in Business:**

• Sentiment analysis from YouTube comments.

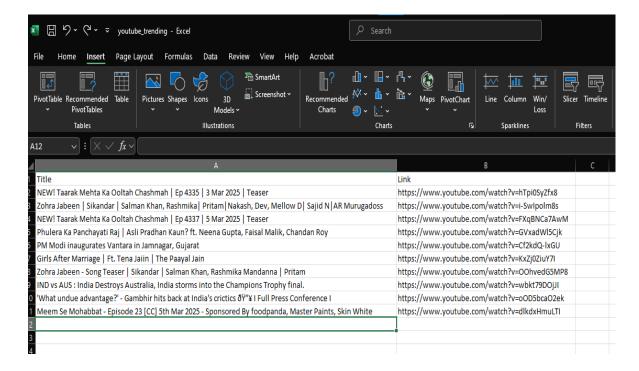
- Competitor analysis by tracking engagement on videos.
- Trend analysis using video views and search patterns.
- Influencer marketing insights.

## Code: -

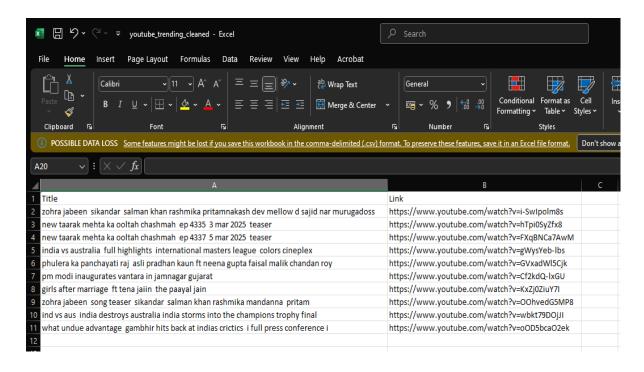
```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
import time
import pandas as pd
#  Automatically install the correct ChromeDriver version
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service)
# Open YouTube Trending Page
driver.get("https://www.youtube.com/feed/trending")
time.sleep(5) # Wait for page to load
# Scrape video titles and links
videos = driver.find_elements(By.XPATH, '//a[@id="video-title"]')
video data = []
for video in videos[:10]: # Get top 10 videos
  title = video.text
  link = video.get attribute("href")
```

```
# Save data to CSV
df = pd.DataFrame(video data)
df.to_csv("youtube_trending.csv", index=False, encoding="utf-8")
print(" Data saved to youtube trending.csv")
# Close browser
driver.quit()
#  Your existing scraping code (unchanged)
# (Scrapes data from YouTube and saves to youtube trending.csv)
import pandas as pd
# Load CSV file
df = pd.read csv("youtube trending.csv")
# Data Cleaning Steps
df = df.drop duplicates() # Remove duplicate videos
df = df.dropna(subset=['Title', 'Link']) # Remove rows with missing values
df['Title'] = df['Title'].str.replace(r'[^\w\s]', ", regex=True) # Remove special characters
df['Title'] = df['Title'].str.lower() # Convert titles to lowercase
# Save cleaned data
df.to csv("youtube trending cleaned.csv", index=False, encoding="utf-8")
print(" Data cleaned and saved to youtube trending cleaned.csv")
```

video data.append({"Title": title, "Link": link})



#### Youtube\_trending csv output



Youtube\_trending\_cleaned csv output